Function Approximation Based Augmented Backstepping Control for an Induction Machine

MOHAMMED BELKHEIRI
University of Amar Thelidji
Department of Electrical Engineering
PO.Box G37, M'kam, 03000 Laghouat
ALGERIA
mbelkhiri@yahoo.com

FARES BOUDJEMA
Ecole Nationale Polytechnique
Process Control Laboratory
PO.Box 162, Elharrach, 16200 Algiers
ALGERIA
fboudjema@yahoo.fr

Abstract: A new control approach is proposed to address the tracking problem of an induction machine based on a modified field-oriented control (FOC) method. In this approach, one relies first on a partially known model to the system to be controlled using a backstepping control strategy. The obtained controller is then augmented by an online neural network that serves as an approximator for the neglected dynamics and modeling errors. The proposed approach is systematic, and exploits the known nonlinear dynamics to derive the stepwise virtual stabilizing control laws. At the final step, an augmented Lyapunov function is introduced to derive the adaptation laws of the network weights. The effectiveness of the proposed controller is demonstrated through computer simulation.

Key–Words: Induction machine, Nonlinear adaptive control, FOC, backstepping, Neural networks, Function Approximation.

1 Introduction

Nonlinear control theory has been applied extensively to the control of induction machines [5],[15],[6],[10],[7] such as nonlinear state feedback control and input-output linearization strategies based on the nonlinear coupled differential equations describing the machine dynamics. Feedback linearization controllers are among the most successful techniques to achieve input/output decoupling, high dynamic performance, and higher power efficiency [17]. The main drawback of such methods is that they rely on a known model of the machine with precise parameters. Backstepping control is applied in the control of induction machines trying to gain from the stabilizing nonliner terms rather than eliminating them in feedback linearization. However, the problem of unknown parameters is usually addressed by authors in literature [3]. Many variants of the developed control laws try to relax the constraints of applying such methods to real systems where the uncertainty is structured and linear in the unknown parameters. Linearly parameterized adaptive control results motivate researchers to find new structures that does not rely on known nonlinearities by using approximation property of neural networks and fuzzy logic approximators. Lewis *et al.* have developed online adaptive control laws with a one hidden layer neural network based on Lyapunov theory for nonlinear systems affine in control [11],[12]. Calise *et al.* [8], proposed a method to relax the affine in control limitation based on the compensation for the inversion error of a feedback linearization controller. The objective of this paper is to develop a control law to control the induction machine that augments a backstepping controller by an artificial neural network which can be extended for a larger class of nonlinear systems with partially known models. In section 2 the problem will be formulated, then field oriented control method is presented in section 3. Backstepping will be applied for the known part of the system in section 4. Neural network augmentation is detailed in section 5. Simulation results are presented in section 6. Section 7 is devoted to some concluding remarks.

2 Induction Machine Modeling

Under the assumptions of linearity of the magnetic circuit and neglecting iron losses, the dynamics of a n_p pole-pair two phase induction motor are given by the following system of differential equations [5]:

$$\left\{ \begin{aligned} u_{sa} &= R_s i_{sa} + L_s \tfrac{di_{sa}}{dt} \\ &\quad + M \tfrac{d}{dt}\left(i_{ra} cos(n_p\theta) - i_{rb} sin(n_p\theta)\right) \\ u_{sb} &= R_s i_{sb} + L_s \tfrac{di_{sb}}{dt} \\ &\quad + M \tfrac{d}{dt}\left(i_{ra} sin(n_p\theta) - i_{rb} cos(n_p\theta)\right) \end{aligned} \right. \tag{1}$$

$$\left\{\begin{array}{l} 0 = R_r i_{ra} + L_r \frac{di_{ra}}{dt} \\ \qquad +M\frac{d}{dt}\left(i_{sa}cos(n_p\theta) + i_{sb}sin(n_p\theta)\right) \\ 0 = R_r i_{rb} + L_r \frac{di_{rb}}{dt} \\ \qquad +M\frac{d}{dt}\left(-i_{sa}sin(n_p\theta) + i_{sb}cos(n_p\theta)\right) \\ J\frac{d\omega}{dt} = n_p M[i_{sb}\left(i_{ra}cos(n_p\theta) - i_{rb}sin(n_p\theta)\right) \\ \qquad -i_{sa}\left(i_{ra}sin(n_p\theta) - i_{rb}cos(n_p\theta)\right)] \end{array}\right. \tag{2}$$

with the flux linkages of the motor phases given by

$$\left\{\begin{array}{l} \lambda_{sa} = L_s i_{sa} + M\left(i_{ra}cos(n_p\theta) - i_{rb}sin(n_p\theta)\right) \\ \lambda_{sb} = L_s i_{sb} + M\left(i_{ra}sin(n_p\theta) - i_{rb}cos(n_p\theta)\right) \\ \lambda_{ra} = L_r i_{ra} + M\left(i_{sa}cos(n_p\theta) + i_{sb}sin(n_p\theta)\right) \\ \lambda_{rb} = L_r i_{rb} + M\left(-i_{sa}sin(n_p\theta) + i_{sb}cos(n_p\theta)\right) \end{array}\right. \tag{3}$$

where $L_s = \frac{\mu_0 \pi l_1 l_2 N_S^2}{8g}, L_r = \frac{\mu_0 \pi l_1 l_2 N_S N_R}{8g}$.

Here N_s and N_R are the number of windings per pole-pair of the stator and rotor phases, respectively. The retarding torque produced by the friction in the ball bearings of the machine is modeled here by $-f\omega$, where f is the viscous friction coefficient .

The control problem is to choose the input stator voltages u_{sa} and u_{sb} in such a way to make the motor angular velocity ω tracks a given reference trajectory and to achieve a given desired torque. The stator currents are usually accessible. whereas, the rotor currents are typically not available for feedback. In fact, the most common type of induction motor is the squirrel cage motor, where rotor currents are distributed on the surface of the rotor making it impractical to measure the current in each rotor bar. The resulting flux can be measured using expensive Hall effect sensors placed in the air gap. The control problem is still difficult due to the coupled complicated model of the system described in equations (1,2). To overcome this problem one could transform it into a simplified form in which $cos(n_p\theta)$ and $sin(n_p\theta)$ expressions are eliminated. Using the following transformation for the flux linkages

$$\left[\begin{array}{c} \psi_{ra} \\ \psi_{rb} \end{array}\right] = \left[\begin{array}{cc} cos(n_p\theta) & -sin(n_p\theta) \\ sin(n_p\theta) & cos(n_p\theta) \end{array}\right] \left[\begin{array}{c} \lambda_{ra} \\ \lambda_{rb} \end{array}\right] \tag{4}$$

Then the dynamic model of the machine in terms of the new state variables $\omega, \psi_{ra}, \psi_{rb}, i_{sa}$, and i_{sb} can be written in this form

$$\left\{\begin{array}{l} \frac{d\omega}{dt} = \mu\left(i_{sb}\psi_{ra} - i_{sa}\psi_{rb}\right) - \frac{f}{J}\omega - \frac{1}{J}\tau_L \\ \frac{d\psi_{ra}}{dt} = -\eta\psi_{ra} - n_p\omega\psi_{rb} + \eta M i_{sa} \\ \frac{d\psi_{rb}}{dt} = -\eta\psi_{rb} + n_p\omega\psi_{ra} + \eta M i_{sb} \\ \frac{i_{sa}}{dt} = \beta\left(\eta\psi_{ra} + n_p\omega\psi_{rb}\right) - \gamma i_{sa} + \frac{1}{\sigma L_s}u_{sa} \\ \frac{i_{sb}}{dt} = \beta\left(\eta\psi_{rb} - n_p\omega\psi_{ra}\right) - \gamma i_{sb} + \frac{1}{\sigma L_s}u_{sb} \end{array}\right. \tag{5}$$

with $\mu = \frac{n_p M}{J L_r}, \sigma = 1 - \frac{M^2}{L_r L_s}, \eta = \frac{R_r}{L_r}, \beta = \frac{M}{\sigma L_r L_s}$, and $\gamma = \frac{M^2 R_r}{\sigma L_r^2 L_s} + \frac{R_r}{\sigma L_s}$.

3 Conventional Field Oriented Control

The key idea of field-oriented control is to transform the system to to another state space representation where the currents regulating the flux and the speed are decoupled [1]. The new coordinate system is a rotating system whose angular position is defined by $\rho = \arctan(\frac{\psi_{rb}}{\psi_{ra}})$. So, instead of working with (ψ_{rb}, ψ_{ra}), one uses the polar coordinate representation (ρ, ψ_d) given by

$$\rho = \arctan(\frac{\psi_{rb}}{\psi_{ra}}), \psi_d = \sqrt{\psi_{ra}^2 + \psi_{rb}^2}. \tag{6}$$

The stator phase currents and voltages are then expressed in this new coordinates as follows

$$\left[\begin{array}{c} i_d \\ i_q \end{array}\right] = \left[\begin{array}{cc} cos(\rho) & sin(\rho) \\ sin(\rho) & cos(\rho) \end{array}\right] \left[\begin{array}{c} i_{sa} \\ i_{sb} \end{array}\right] \tag{7}$$

$$\left[\begin{array}{c} u_d \\ u_q \end{array}\right] = \left[\begin{array}{cc} cos(\rho) & sin(\rho) \\ sin(\rho) & cos(\rho) \end{array}\right] \left[\begin{array}{c} u_{sa} \\ u_{sb} \end{array}\right] \tag{8}$$

The electromagnetic dynamic model of the induction motor in the fixed stator (direct and quadrant) d-q reference frame can be developed yielding

$$\left\{\begin{array}{l} \frac{d\omega}{dt} = \mu\psi_d i_q - \frac{f}{J}\omega - \frac{1}{J}\tau_L \\ \frac{d\psi_d}{dt} = -\eta\psi_d + \eta M i_d \\ \frac{i_d}{dt} = -\gamma i_d + \beta\eta\psi_d + n_p\omega i_q + \eta M i_q^2/\psi_d + \frac{1}{\sigma L_s}u_d \\ \frac{i_q}{dt} = -\gamma i_q - \beta n_p\omega\psi_d - n_p\omega i_d - \eta M i_d i_q/\psi_d + \frac{1}{\sigma L_s}u_q \\ \frac{d\rho}{dt} = n_p\omega + \eta M i_q/\psi_d \end{array}\right. \tag{9}$$

Note that the electromagnetic torque $\tau_e = J\mu\psi_d i_q$ is now just proportional to the product of two state variables ψ_d and i_q. Furthermore, applying the nonlinear state feedback control for system (9)

$$\left[\begin{array}{c} u_d \\ u_q \end{array}\right] = \sigma L_s \left[\begin{array}{c} -\beta\eta\psi_d - n_p\omega i_q - \eta M i_q^2/\psi_d + \bar{u_d} \\ \beta n_p\omega\psi_d + n_p\omega i_d + \eta M i_d i_q/\psi_d + \bar{u_q} \end{array}\right] \tag{10}$$

Then the closed loop system is obtained as follows:

$$\left\{\begin{array}{l} \frac{d\omega}{dt} = \mu\psi_d i_q - \frac{f}{J}\omega - \frac{1}{J}\tau_L \\ \frac{d\psi_d}{dt} = -\eta\psi_d + \eta M i_d \\ \frac{i_d}{dt} = -\gamma i_d + \bar{u_d} \\ \frac{i_q}{dt} = -\gamma i_q + \bar{u_q} \\ \frac{d\rho}{dt} = n_p\omega + \eta M i_q/\psi_d \end{array}\right. \tag{11}$$

From (11), it is clear after field-oriented control and nonlinear state feedback, the final closed-loop system has a simpler structure. Moreover, the flux amplitude dynamics depend only on the direct current i_d and the direct voltage u_d. Thus, it can be regulated to achieve a given flux amplitude that can generate the desired electromagnetic torque τ. However, the robustness to parameter variation of field orientation and nonlinear state feedback control cannot be guaranteed since the designed controller relies entirely on the exact values of the induction motor parameters which are usually estimated so any mismatch could cause system instability and the desired objective is no longer achieved.

4 Backstepping Control

Backstepping is a constructive tool for nonlinear systems tries to find a stabilizing control law along a Lyapunov function ensuring closed loop system stability. In this section, it is applied for a known model of the Induction Machine. Let $\tau_e^* = J\mu\psi_d^* i_q$ be the desired torque to be generated with the corresponding reference flux ψ_d^*. Defining the flux and the speed errors as:

$$\begin{bmatrix} \tilde{\psi_d} \\ \tilde{\omega} \end{bmatrix} = \begin{bmatrix} \psi_d - \psi_d^* \\ \omega - \omega^* \end{bmatrix} \tag{12}$$

Differentiation of equation (12) yields to the following error dynamics

$$\begin{cases} \dot{\tilde{\psi_d}} = -\eta(\tilde{\psi_d} + \psi_d^*) + \eta M i_d - \dot{\psi}_d^* \\ \dot{\tilde{\omega}} = \mu\psi_d i_q - \frac{f}{J}(\tilde{\omega} + \omega^*) - \frac{1}{J}\tau_L - \dot{\omega}^* \end{cases} \tag{13}$$

4.1 Step 1

Let us define the first candidate Lyapunov function:

$$V_1 = \frac{1}{2}\left[\tilde{\psi_d}^2 + \tilde{\omega}^2\right] \tag{14}$$

For which the time derivative is expressed as:

$$\dot{V}_1 = \tilde{\psi_d}\dot{\tilde{\psi_d}} \qquad + \tilde{\omega}\dot{\tilde{\omega}} \tag{15}$$

Substituting for the error dynamics in (15) yields

$$\begin{aligned} \dot{V}_1 = \tilde{\psi_d}&\left[-\eta(\tilde{\psi_d} + \psi_d^*) + \eta M i_d - \dot{\psi}_d^*\right] \\ &+\tilde{\omega}\left[\mu\psi_d i_q - \frac{f}{J}(\tilde{\omega} + \omega^*) - \frac{1}{J}\tau_L - \dot{\omega}^*\right] \end{aligned} \tag{16}$$

To render it negative definite, we may choose the fictitious control signals, the direct and quadratic currents i_d and i_q as

$$\begin{cases} i_d^* = \frac{1}{\eta M}\left(\eta\psi_d^* + \dot{\psi}_d^*\right) - k_1\tilde{\psi_d} \\ i_q^* = \frac{1}{\mu\psi_d}\left(\frac{f}{J}\omega^* + \frac{1}{J}\tau_L + \dot{\omega}^* - k_2\tilde{\omega}\right) \end{cases} \tag{17}$$

with k_1 and k_2 are positive design parameters ensuring that the tracking error dynamics will converge exponentially to zero.

4.2 Step 2

Let $\tilde{i_d}$ and $\tilde{i_q}$ be the direct and quadratic current errors defined as:

$$\begin{bmatrix} \tilde{i}_d \\ \tilde{i}_q \end{bmatrix} = \begin{bmatrix} i_d - i_d^* \\ i_q - i_q^* \end{bmatrix} \tag{18}$$

The augmented error dynamics become

$$\begin{cases} \dot{\tilde{\psi_d}} = -\eta(1 + Mk_1)\tilde{\psi_d} + \eta M\tilde{i}_d \\ \dot{\tilde{i}}_d = -\gamma\tilde{i}_d + \bar{u}_d - \gamma i_d^* - i_d'^* \\ \dot{\tilde{\omega}} = -(\frac{f}{J} + k_2)\tilde{\omega} + \mu\psi_d\tilde{i}_q \\ \dot{\tilde{i}}_q = -\gamma\tilde{i}_q + \bar{u}_q - \gamma i_q^* - i_q'^* \end{cases} \tag{19}$$

Defining the fictitious control signals

$$\begin{cases} v_d = \bar{u}_d - \gamma i_d^* - i_d'^* \\ v_q = \bar{u}_q - \gamma i_q^* - i_q'^* \end{cases} \tag{20}$$

The second step of backstepping suggests the following Lyapunov candidate

$$V_2 = V_1 + \frac{1}{2}\left[\tilde{i_d}^2 + \tilde{i_q}^2\right] \tag{21}$$

The derivative of V_2 along the augmented error dynamics (19) is given by

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + \tilde{i_d}\dot{\tilde{i_d}} + \tilde{i_q}\dot{\tilde{i_q}} \\ \dot{V}_2 &= -\eta(1 + Mk_1)\tilde{\psi_d}^2 + \tilde{i_d}\left(\eta M\tilde{psi}_d - \gamma\tilde{i}_d + v_d\right) \\ &\quad -(\tfrac{f}{J} + k_2)\tilde{\omega}^2 + \tilde{i_q}\left(\mu\psi_d\tilde{\omega} - \gamma\tilde{i}_q + v_q\right) \end{aligned} \tag{22}$$

This is the last step of standard backstepping where the control signals appear and could be chosen in a manner to render the derivative of the Lyapunov candidate negative definite as follows:

$$\begin{cases} v_d = -\eta M\tilde{psi}_d - k_3\tilde{i}_d \\ v_q = -\mu\psi_d\tilde{\omega} - k_4\tilde{i}_q \end{cases} \tag{23}$$

with k_3 and k_4 are positive parameters selected to ensure that the current dynamics converge faster than those of the speed and flux.

The obtained control law is implemented given all the state variables are available for feedback and the induction motor parameters are known exactly. In the next section we will propose a control technique to make our designed controller robust and we will use instead of the missed states their estimates.

5 Error Approximation Augmented Control

The control law given in equation (24) can be implemented if these conditions are fulfilled:

- All the state are accessible for feedback, and
- All the nonlinear functions are precisely known.

To relax the first two conditions we can use the same idea as in [9] by introducing a function approximator structure that can overcome the mismatch between IM Model used in control design and the real system. This approximator can be an Artificial neural network, a fuzzy logic approximator or any other structure. To derive the adaptation rules for the approximator, we can proceed further byaugmenting the Lyapunov function found in the last step of backstepping .

According to equations (10,15 and 22) the control signals depend on all the system parameters, assuming that for each parameter ϱ we have available an estimate $\hat{\varrho}$. The control strategy will be changed by adding an adaptive signal which is the output of a function approximation structure in equation (24) which becomes:

$$\left\{ \begin{array}{l} v_d = -\eta M p\tilde{s}i_d - k_3 \tilde{i}_d + u_d^a \\ v_q = -\mu \psi_d \tilde{\omega} - k_4 \tilde{i}_q + u_q^a \end{array} \right. \tag{24}$$

Substituting for v_d, and v_q in system equations (19) and assuming that all neglected terms for each subsystem as an error signal e_i. The extended error dynamics take the form:

$$\left\{ \begin{array}{l} \dot{\tilde{\psi}}_d = -c_1 \tilde{\psi}_d + \eta M \tilde{i}_d \\ \dot{\tilde{i}}_d = -c_3 \tilde{i}_d - \eta M \tilde{\psi}_d + e_d - u_d^a \end{array} \right. \atop \left\{ \begin{array}{l} \dot{\tilde{\omega}} = -c_2 \tilde{\omega} + \mu \psi_d \tilde{i}_q \\ \dot{\tilde{i}}_q = -c_4 \tilde{i}_q - \mu \psi_d \tilde{\omega} + e_q - u_q^a \end{array} \right. \tag{25}$$

The two subsystems are almost identical, for the rest of the paper we will consider only the dynamics of the flux subsystem to simplify writing:

$$\left\{ \begin{array}{l} \dot{\tilde{\psi}}_d = -c_1 \tilde{\psi}_d + \eta M \tilde{i}_d \\ \dot{\tilde{i}}_d = -c_3 \tilde{i}_d - \eta M \tilde{\psi}_d + e_d - u_d^a \end{array} \right. \tag{26}$$

Usually in the literature, neural networks are used in function approximation in the modeling phase [19], however we will use them in controller design to cancel the effect of uncertainty due to neglected dynamics and unknown or varying system parameters,given some conditions are fulfilled.

Assume that there exists an artificial neural network as shown in fig. 1 that approximates the neglected dynamics e_d.

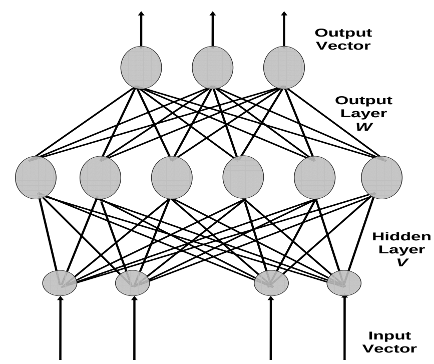

Figure 1: A Neural Network with two layers

The approximated error signal can be expressed as:

$$e_d = W^{\mathrm{T}} \Phi(V, \mu) + \varepsilon(\mu) \; \forall \mu \in D \tag{27}$$

where

$V \in D_V \subset \Re^{N_1}$ is the hidden layer weight vector

$W \in D_W \subset \Re^{N_2}$ is the output layer weight vector

Φ is a set of basis functions

μ is the network input vector, and

ε is the neural network reconstruction error

Assuming that the approximation reconstruction error is bounded on some domain D by $\|\varepsilon(\mu)\| \leq \epsilon^*$, $\forall \mu \in D$.

The weight vectors are assumed to be bounded $\|W\| \leq W^*$ and their adjustment can be done online. However the whole controller implementation will be difficult as the number of the parameters increases for complex problems. So the first layer parameters can be adjusted off-line in such a way $\Phi(\mathbf{V}, \mu)$ is a basis [19] whereas the adaptation law for the second layer is included in the controller design.

5.1 Neural Network Inner Weight Vector Adaptation

Substituting for e_d in equation (26) yields

$$\begin{cases} \dot{\tilde{\psi}}_d = -c_1\tilde{\psi}_d + \eta M\tilde{i}_d \\ \dot{\tilde{i}}_d = -c_3\tilde{i}_d - \eta M\tilde{\psi}_d + W^{\mathrm{T}}\Phi(V,\mu) + \varepsilon(\mu) - u_d^a \end{cases} \tag{28}$$

The adaptive control signal u_d^a is designed to cancel the effect of unknown nonlinear terms and is chosen to take the following form.

$$u_d^a = \hat{W}^{\mathrm{T}}\Phi(\mathbf{V},\mu) \tag{29}$$

Since the inner layer weights $\mathbf{V}$ are adjusted offline to fit the real weights V then we can write the mismatch between the adaptive signal and the real neural network as:

$$\begin{aligned} e_d - u_d^a &= W^{\mathrm{T}}\Phi(V,\mu) - \hat{W}^{\mathrm{T}}\Phi(\mathbf{V},\mu) + \varepsilon(\mu) \\ &= W^{\mathrm{T}}(\Phi(\mathbf{V},\mu) + e_v) - \hat{W}^{\mathrm{T}}\Phi(\mathbf{V},\mu) + \varepsilon(\mu) \\ &= \left(W^{\mathrm{T}} - \hat{W}^{\mathrm{T}}\right)\Phi(\mathbf{V},\mu) + \delta \end{aligned} \tag{30}$$

where $\delta \leq \overline{\delta}$ is a bounded signal represents the reconstruction error of the neural network plus the error caused by the inner weights mismatch.

Let $\tilde{W} = W - \hat{W}$ the error between the real neural network weight and its estimate.

Thus equation (28) can be rewritten as

$$\begin{cases} \dot{\tilde{\psi}}_d = -c_1\tilde{\psi}_d + \eta M\tilde{i}_d \\ \dot{\tilde{i}}_d = -c_3\tilde{i}_d - \eta M\tilde{\psi}_d + \tilde{W}^{\mathrm{T}}\Phi(\mathbf{V},\mu) + \delta \end{cases} \tag{31}$$

To get the adaptation law for the network weights, We suggest the augmented Lyapunov function

$$V_d^a = \frac{1}{2}\left(\tilde{\psi_d}^2 + \tilde{i}_d^2 + \tilde{W}^{\mathrm{T}}F^{-1}\tilde{W}\right) \tag{32}$$

where $F > 0$ is an adaptation gain. Differentiating V_d^a with respect to the error dynamics in equation(31) yields

$$\dot{V}_d^a = -c_1\tilde{\psi_d}^2 - c_3\tilde{i_d}^2 + \tilde{W}^{\mathrm{T}}F^{-1}\dot{\tilde{W}} + \tilde{i}_d\left[\tilde{W}^{\mathrm{T}}\Phi(\mathbf{V},\mu) + \delta\right] \tag{33}$$

Using the fact that $\dot{\tilde{W}} = -\dot{\hat{W}}$, equation (33) can be written as

$$\dot{V}_d^a = -c_1\tilde{\psi_d}^2 - c_3\tilde{i_d}^2 - \tilde{W}^{\mathrm{T}}F^{-1}\dot{\hat{W}} + \tilde{i}_d\left[\tilde{W}^{\mathrm{T}}\Phi(\mathbf{V},\mu) + \delta\right] \tag{34}$$

Hence, we can derive this adaptive law for the network parameters W which will make the derivative augmented Lyapunov function to be negative definite in some predefined domain.

$$\dot{\hat{W}} = -F\left[\Phi(\mathbf{V},\mu)\tilde{i}_d + 2G(\hat{W} - W_0)\right] \tag{35}$$

It can be proven that the adaptive law (35) ensures that $\dot{V}_d^a \leq -\underline{N}$. Which is negative definite in some domain.

Substituting (35) in (34) yields

$$\begin{aligned} \dot{V}_d^a = &-c_1\tilde{\psi_d}^2 - c_3\tilde{i_d}^2 - \tilde{W}^{\mathrm{T}}\left[\Phi(\mathrm{V},\mu)\tilde{i_d} + 2G(\hat{W} - W_0)\right] \\ &+\tilde{i_d}\left(\tilde{W}^{\mathrm{T}}\Phi(\mathrm{V},\mu) - \varepsilon(\mu)\right) \end{aligned} \tag{36}$$

or

$$\dot{V}_a = -c_1\tilde{\psi_d}^2 - c_3\tilde{i_d}^2 - 2\tilde{W}^{\mathrm{T}}G(\hat{W} - W_0) - \tilde{i_d}\varepsilon(\mu) \tag{37}$$

Using the fact that the neural networks weight vector is bounded, the derivative of the augmented Lyapunov candidate can be upper bounded as

$$\begin{aligned} \dot{V}_a \leq &-c_1\tilde{\psi_d}^2 - c_3\tilde{i_d}^2 + |\tilde{i_d}|\epsilon^* \\ &-G\|\tilde{W}\|^2 - G\|\hat{W} - W_0\|^2 + G\|W - W_0\|^2 \end{aligned} \tag{38}$$

Putting $\underline{U} = c_1\tilde{\psi_d}^2 + G\|\hat{W} - W_0\|^2$ yields

$$\begin{aligned} \dot{V}_a \leq &-\underline{U} - c_3\tilde{i_d}^2 + |\tilde{i_d}|\epsilon^* \\ &-G\|\tilde{W}\|^2 + G\|W - W_0\|^2 \end{aligned} \tag{39}$$

Completing the squares using $\epsilon^*|\tilde{i_d}| = -\frac{1}{2}\left(\epsilon^* - |\tilde{i_d}|\right)^2 + \frac{1}{2}\epsilon^{*2} + \frac{1}{2}\tilde{i_d}^2$, we get

$$\begin{aligned} \dot{V}_a \leq &-\underline{U} - c_3\tilde{i_d}^2 - G\|\tilde{W}\|^2 + G\|W - W_0\|^2 \\ &-\tfrac{1}{2}\left(\epsilon^* - |\tilde{i_d}|\right)^2 + \tfrac{1}{2}\epsilon^{*2} + \tfrac{1}{2}\tilde{i_d}^2 \end{aligned} \tag{40}$$

Further, it can be written as

$$\begin{aligned} \dot{V}_a \leq &-\underline{U} - (c_3 - \tfrac{1}{2})\tilde{i_d}^2 - G\|\tilde{W}\|^2 - \tfrac{1}{2}\left(\epsilon^* - |\tilde{i_d}|\right)^2 \\ &+G\|W - W_0\|^2 + \tfrac{1}{2}\epsilon^{*2} \end{aligned} \tag{41}$$

The following conditions

$$\begin{aligned} |\tilde{i_d}| &> \sqrt{\tfrac{2\epsilon^{*2} + 2G\|W - W_0\|^2}{2c_3 - 1}} \\ \|\tilde{W}\|^2 &> \sqrt{\tfrac{\epsilon^{*2} + G\|W - W_0\|^2}{G}} \end{aligned} \tag{42}$$

with $c_3 > \frac{1}{2}$, and $G > 0$ ensures that $\dot{V}_a \leq -\underline{U}$. Which is negative definite in some domain defined by conditions in (42).

The resulting controller looks like the controller proposed in [9] however here it is obtained in a constructive manner for a triangular system without adding an SPR condition. It can be implemented and we have a freedom in selecting the linear controller gains c_i to achieve a desirable performance.

5.2 Hidden Layer Weight Off-Line Selection

As stated earlier, we have many choices in designing the hidden layer weights depending on the used type of neural network. In this note we will show a design alternative based on Radial Basis Function Neural Networks (RBFNN), which are highly recommended for function approximation due to their simple training.

An RBFNN is composed of two layers; a hidden layer contains a set of neurons with their associated centers, the output of each neuron gives the distance between the input vector μ and its center ν_i. The output ψ of the network is a linear combination of the outputs of the hidden layer as.

$$\psi(\mu_j) = \sum_{i=1}^{N_1} w_i \varphi_{ij} \tag{43}$$

where $\varphi_{ij} = exp\left(\frac{\|\mu_j - \nu_i\|}{\rho}\right)$ is a Gaussian activation function of the distance between the input μ_j and the i^{th} center ν_i.

Defining $W = \begin{bmatrix} w_1 & \cdots & w_{N1} \end{bmatrix}$, $V = \begin{bmatrix} \nu_1 & \cdots & \nu_{N1} \end{bmatrix}$. The design of an RBF network to approximate a given function consists of selecting V in such a way that we construct a set of basis functions [13] and W is adjusted online using an adaptive law to be illustrated in the next subsection. Therefore we have an identification problem based on model (15) which includes the selection of $N1$ model terms $V = \begin{bmatrix} \nu_1 & \cdots & \nu_{N1} \end{bmatrix}$ from a full model set of $M > N1$ terms $U = \begin{bmatrix} \mu_1 & \cdots & \mu_M \end{bmatrix}$ (typically hundreds or even thousands of terms) while μ_i is defined earlier as a tapped delay line of possible values of the input/output and the estimate of z.

We construct the regression matrix Φ_L corresponding to the set of the input vectors U_L a subset of the starting set of centers U.

$$\Phi_L = \begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1L} \\ \vdots & \ddots & \vdots \\ \varphi_{L1} & \cdots & \varphi_{LL} \end{bmatrix}. \tag{44}$$

Notice that this matrix is symmetric and all the diagonal elements are ones.

It has been shown that the orthogonal algorithm can be employed in selecting the optimal model structure V and to estimate the parameters simultaneously [4]. The orthogonal term selection is formulated using the error reduction ratio vector $ERR_L = \begin{bmatrix} err_i & \cdots & err_L \end{bmatrix}$ defined by:

$$ERR_L = \frac{W^{\mathrm{T}} \Phi_L^2 W}{\Psi^{\mathrm{T}} \Psi} \tag{45}$$

To find $N1$ optimal model terms V a stepwise approach is applied to the full model set U. At each step, the model term with the maximum err_j value from all of the model terms excluding the previously selected terms is chosen. The selection is terminated at the $N1^{th}$ step where a desired tolerated error tol is reached.

$$1 - \sum_{k=1}^{N1} err_k < tol \tag{46}$$

6 Simulations and Results

In this section we will investigate the performance of the proposed control strategy to an induction motor. The physical and electrical parameters of the machine under investigation are summarized in table 1.

Table 1: Induction Machine Simulation Parameters

Parameter	Value
L_r	0.094 mH
L_s	105 mH
R_s	1.47 Ω
R_r	0.79 Ω
p	2
J	0.0077 Kgm^2
f	0.0029 Kgm^2/s

The tracking performance of speed and flux is shown in the following figures: 1- The ideal case (exact model fig.1), 2- Figures (2, 4) show the poor performance in the presence of uncertainty $(+/-50\%)$ of parameter variation without neural network augmentation. Figures (3,5) show the tracking performance after neural network augmentation, It is clear that the neural network has compensated for the unknown terms.

Simulations show that this method is robust for parameter variation.

7 Conclusion

A new robust adaptive nonlinear controller is proposed in this paper to address the tracking problem for a two phase induction machine based on a modified version of FOC. In this scheme, the stability is ensured for the closed loop system and all the errors (tracking and neural network weights) tend asymptotically to zero. The RBF center selection is done using orthogonal least squares structure selection method to minimize the number of parameters of the resulting controller. Finally simulations are presented to highlight the achieved performance although we have assumed a partly known model of the system to be con-

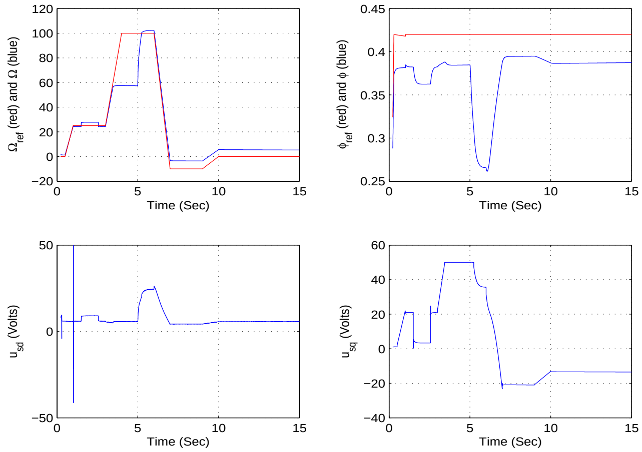

Figure 2: Simulation without uncertainty but unknown load.

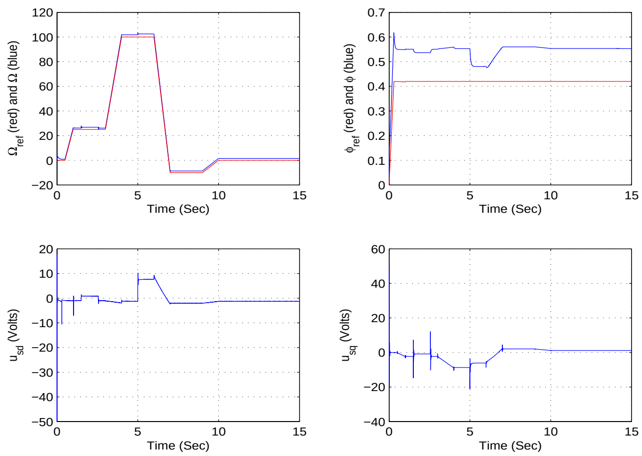

Figure 3: simulation with +50 % uncertainty but without ANN.

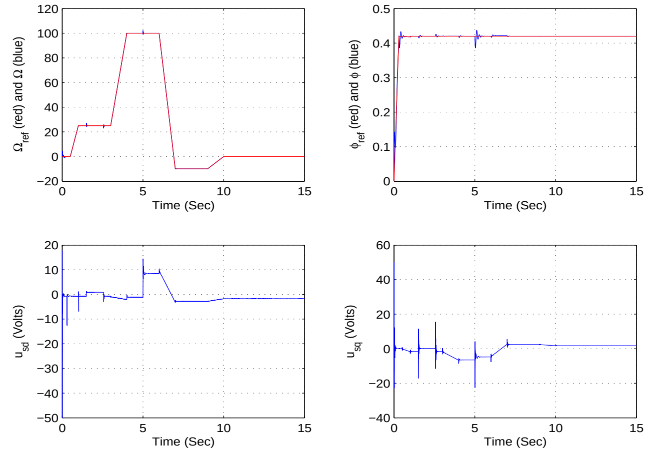

Figure 4: Simulation with $+50\%$ uncertainty with ANN

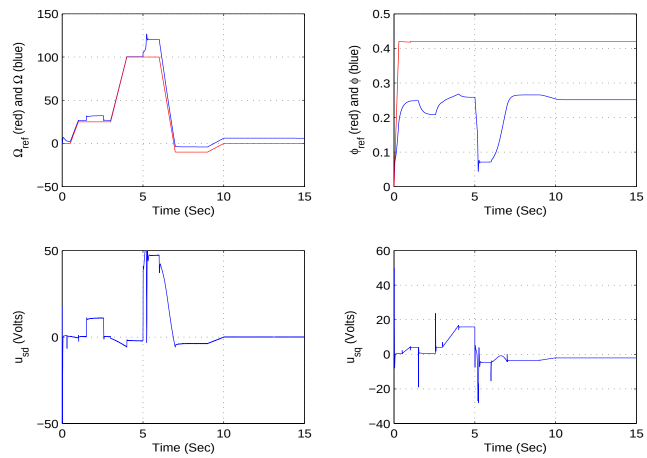

Figure 5: Simulation with -50% uncertainty but without ANN

trolled. As a future research we will propose to add other neural network structures for sensorless control and we encourage researchers to try other artificial intelligence techniques as function approxmators and genetic algorithms (GA) to optimize the function approximators structure.

References:

[1] A. Razzouk, A. Cheriti, J. Olivier, and P. Sicard, Field Oriented Control for Induction Motors Using Neural Networks Decouplers, *proceding of IDS conference*, Canada 1995.

[2] H. Shieh, and K. Shyu, Nonlinear Sliding-Mode Torque Control with Adaptive Backstepping Approach for Induction Motor Drive, *IEEE Transactions on Industrial Electronics*, vol. 46, pp. 380-389, Apr. 1999.

[3] M. Kristic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, 1995.

[4] M. Korenberg and L. Paarmaan, Orthogonal Parameter Estimation Algorithm for Nonlinear Stochastic Systems, *Int.Journal of Control*, vol. 48, 1988.

[5] J. Chiasson, *Modeling and High Performance Control of Electric Machines,* John Wiley & Sons, New Jersey, 2005.

[6] J. Chiasson, Dynamic feedback linearization of the induction motor, *IEEE Transactions on Automatic Control*, vol. 38, pp. 1588-1594, Oct. 1993.

[7] D. Taylor, Nonlinear control of electric machines: an overview, *IEEE Control System Magazine,* vol. 14, pp. 41-51, Dec. 1994.

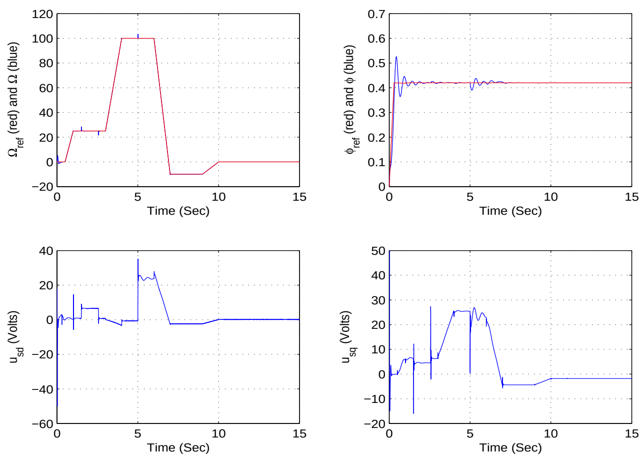

Figure 6: Simulation with -50% uncertainty with ANN

[8] A. Calise and R. Rysdyk, Nonlinear Adaptive Flight Control Using Neural Networks, *IEEE Control System Magazine*, vol. 18, no. 6, 1998.

[9] N. Hovakimyan, A. Calise and N. Kim, Adaptive Output Feedback Control of a Class of Multi-Input Multi-Output Systems Using Neural Networks, *Int. Journal of Control*, vol. 77, no. 15, 2004.

[10] A. Onea, V. Horrga, and M. Ratoi, Indirect Vector Control of Induction Motor, *Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization,* Lisbon,Portugal, September 22-24, 2006.

[11] F. Lewis, A. Yesildirek, and K. Liu, Multi-Layer Neural-Net Robot Controller with Guaranteed Tracking Performance, *IEEE Trans. Neural Networks*, vol. 7, no.2, 1996.

[12] F. Lewis, A. Yesildirek, and K. Liu, Neural Net Robot Controller: Structure and Stability Proofs, *Journal of Intelligent and Robotic System*, vol. 12, 1996.

[13] L. Ljung, *System identification : Theory for the User,* Englewood cliffs, New Jersey, Prentice Hall 1987.

[14] S. Sastry and A. Isidori, Adaptive Control of Linearizable Systems, *IEEE Trans. Autom. Control,* vol. 34, no. 11, 1989.

[15] I. Siller-Alcala, M. Fichouch, J. Jaimes-Ponce, and R. Alcanta-Ramirez, Position Predictive Control for an Induction Motor, *Proceedings of the 3rd WSEAS/IASME International Conference on Dynamical Systems and Control,* Arcachon, France, October 13-15, 2007.

[16] R. R. Selmic and F. L. Lewis, Identification of Nonlinear Systems Using RBF Neural Networks: Application to Multimodel Failure Detection, *XVI Intern. Conf. on Material Flow, Mach. And Dev. in Industry*, University of Belgrade, December 2000.

[17] J. Slotine and W. Li, *Applied Nonlinear Control,* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[18] A. Yesildirek and F. Lewis, Feedback Linearization using Neural Networks, *Automatica*, vol. 31, no. 11, 1995.

[19] W. Yu and X. Li, Some New Results on System Identification with Dynamical Neural Networks, *IEEE Trans. on Neural Networks*, vol. 12 no. 2, March 2001.